

10.1.2 设备控制器 (I/O 接口)

输入输出接口的基本功能是:

1. 实现主机和外围设备之间的通信联络控制, 其中包括同步控制、设备选择和中断控制等。DMA 设备还应具有直接访问存储器功能, 并给出存储器地址。

CPU 发 I/O 数据传送命令时, 必须指明设备地址码, 经各设备接口译码后, 让选中设备参加数据传送。

实现数据缓冲, 以达到主机同外围设备之间的速度匹配。

在接口电路中, 一般设置一个或几个数据缓冲寄存器。在传送过程中, 先将数据送入数据缓冲寄存器, 然后再送到目的设备(输出)或主机(输入)。

接受主机的命令, 提供设备接口的状态, 并按照主机的命令控制设备。

输入输出接口类型有:

(1) 按照数据传送的宽度可分为并行接口和串行接口。在并行接口中, 设备和接口是将一个字节(或字)的所有位同时传送。在串行接口中, 设备和接口间的数据是一位一位串行传送的, 而接口和主机之间是按字节或字并行传送。接口要完成数据格式的串-并变换。在 10.6 节将举例说明。

(2) 按照数据传送的控制方式可分成程序控制输入输出接口、程序中断输入输出接口和直接存储器存取 (DMA) 接口等。

10.1.3 I/O 数据传送控制方式

一般把 I/O 数据传送控制方式分为五种。

1. 程序直接控制方式

程序直接控制方式 (Programmed Direct Control) 就是完全通过程序来控制主机和外围设备之间的信息传送。通常的办法是在用户的程序中安排一段由输入输出指令和其它指令所组成的程序段直接控制外围设备的工作。

一般采用条件传送或称状态驱动方式。传送时, 首先启动设备, 发出启动命令 (START)。接着 CPU 等待外围设备完成接收或发送数据的准备工作。在等待时间内, CPU 不断地用一条测试指令检测外围设备工作状态标志触发器。一旦测试到标志触发器已置成“完成”状态, 即可进行数据传送。这种控制方式虽简单, 但 CPU 和外围设备只能串行工作, 而 CPU 的速度比 I/O 设备的速度快得多, 所以 CPU 的大量时间都处于等待、空闲状态, 使系统效率大大降低, 实际上这种方法极少采用。

2. 程序中断传送方式

在程序中断传送 (Program Interrupt Transfer) 方式中, 通常在程序中安排一条指令, 发出 START 信号启动外围设备, 然后机器继续执行程序。当外围设备完成数据传送的准备后, 便向 CPU 发“中断请求 (INT) 信号。CPU 接到请求后可以停止正在运行的程序, 则在一条指令执行完后, 转去执行“中断服务程序”, 完成传送数据工作, 通常传送一个字或一个字节。传送完毕仍返回原来的程序。

由于系统在启动外围设备后到数据的准备完成这段时间内一直在执行原程序, 不是

处于踏步等待状态,而仅仅在外围设备交换数据的准备工作完成之后才中止程序的继续执行,转而进行数据传送。因此,这在一定程度上实现了 CPU 和外围设备的并行工作。此外,有多台外设依次启动后,可同时进行数据交换的准备工作。若在某一时刻有几台外围设备发出中断请求信号,CPU 可根据预先规定好的优先顺序,按轻重缓急去处理几台外设的数据传送,从而实现了外围设备的并行工作。因此,程序中断方式大大提高了计算机系统的工作效率。

但对于一些工作频率较高的外设,例如,磁盘、磁带等,数据交换是成批的,且单位数据之间的时间间隔较短,如果也采用程序中断方式,将造成信息丢失。因此,对于面向信息块的设备的数据传送采取了下述传送方式。

3. 直接存储器存取方式

直接存储器存取 (Direct Memory Access——DMA) 方式的基本思想是在外围设备和主存之间开辟直接的数据交换通路。在正常工作时,所有工作周期均用于执行 CPU 的程序。当外围设备将要输入或输出的数据准备好后,占用总线一个工作周期,和主存直接交换数据。这个周期过后,CPU 又继续控制总线,执行原程序。如此重复,直到整个数据块的数据传送完毕。这项工作是由 I/O 系统中增设的 DMA 控制器完成的,由它给出每次传送数据的主存地址,并统计已传送数据的个数以确定是否传送结束。除了在数据块传送的起始和结束时需用中断分别进行前处理和后处理外,无需 CPU 的频繁干预。主存储器被并行工作的 CPU 和 I/O 子系统所共享。

DMA 方式也存在着局限性。首先,对外围设备的管理和某些操作的控制仍需由 CPU 承担。在大中型计算机系统中,系统所配备的外设种类多、数量大,这样,对外设的管理和控制也就愈来愈多,愈来愈复杂。大容量外存的使用,使主存和外存之间的调度极为频繁,数据流量大幅度增加,有时还要求多个 DMA 同时使用,引起访问主存的冲突增加。因此,在大型计算机系统中通常设置专门的硬件装置——通道。

4. I/O 通道控制方式 (I/O Channel Control)

“通道”不是一般概念的 I/O 通路,它是一个专用的名称。通道能独立地执行用通道命令编写的输入输出控制程序,产生相应的控制信号送给由它管理的设备控制器,继而完成复杂的输入输出过程。通道是一种通用性和综合性都较强的输入输出方式,它代表了现代计算机组织向功能分布方向发展的初始发展阶段,形成了如图 10.1 的结构。

I/O 通道具有自己的指令系统,并能实现指令所控制的操作,所以,I/O 通道已具备处理机的功能。但它仅仅是面向外围设备的控制和数据传送,其指令系统也仅仅是几条简单的与 I/O 操作有关的命令。它要在 CPU 的 I/O 指令指挥下启动、停止或改变工作状态。在 I/O 处理过程中,有一些操作,如码制转换、数据块的错误检测与校正,一般仍由 CPU 来完成。因此,I/O 通道不是一个完全独立的处理机,它只是从属于 CPU 的一个专用 I/O 处理器。它的进一步发展是引入专用的输入输出处理机。

5. 外围处理机方式

输入输出处理机通常称作外围处理机 (Peripheral Processor Unit——PPU)。这种外围处理机的结构更接近一般处理机,甚至就是一般小型通用计算机。它可完成 I/O 通道所要完成的 I/O 控制,还可完成码制变换、格式处理、数据块的检错、纠错等操作。它可具

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁷

H04L 12/413

G06F 13/38 H04Q 7/32

G06F 13/42

[12] 发明专利申请公开说明书

[21] 申请号 99815656.6

[43] 公开日 2002 年 1 月 30 日

[11] 公开号 CN 1333964A

[22] 申请日 1999.12.13 [21] 申请号 99815656.6

[30] 优先权

[32] 1999.1.15 [33] US [31] 09/232,291

[86] 国际申请 PCT/SE99/02329 1999.12.13

[87] 国际公布 WO00/42740 英 2000.7.20

[85] 进入国家阶段日期 2001.7.16

[71] 申请人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72] 发明人 A·库拉 J·乌格马克

U·比约肯格伦

[74] 专利代理机构 中国专利代理(香港)有限公司

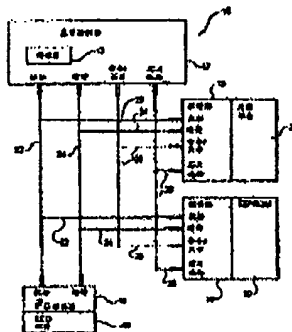
代理人 程天正 李亚非

权利要求书 3 页 说明书 7 页 附图页数 3 页

[54] 发明名称 接口交错

[57] 摘要

用于在集成电路(12)与第一设备(14)和第二设备(18)中每个设备之间传送数据的方法和系统。在该集成电路和该第一设备之间的通信使用第一接口协议,而在该集成电路和该第二设备之间的通信使用第二接口协议。但是这两种接口协议都共享同一条数据总线(22)和时钟总线(24)。为了使用第二接口协议通信,首先通过经共享数据总线发送未标识该使用第一接口协议的第一设备的地址数据而令该第一设备去激活,由此使得该第一设备进入非活动状态,直到在该共享数据总线(22)上检测到结束条件为止。接着,假使在该共享数据总线(22)上没有无意中生成用于第一接口协议的开始或结束条件,则可以执行使用第二接口协议的通信。



ISSN 1008-4274

知识产权出版社出版

01.07.16

权 利 要 求 书

1. 用于在集成电路和多个其它设备之间传送数据的方法，其中该集成电路使用第一接口协议与第一设备通信以及使用第二接口协议与第二设备通信，包括的步骤有：

- 5 使用第一接口协议通过共享数据总线在集成电路和第一设备之间间歇地传输数据，该共享数据总线连接到集成电路、第一设备和第二设备；

 令使用第一接口协议的通信去激活；

- 10 当使用第一接口协议的通信去激活时，允许使用第二接口协议的通信；以及

 使用第二接口协议通过共享数据总线在集成电路和第二设备之间传输数据，其中在不重新激活使用第一接口协议的通信的情况下可执行使用第二接口协议传输数据的步骤。

- 15 2. 权利要求 1 的方法，其中第一接口协议包括智能互连通信 (I²C)。

 3. 权利要求 2 的方法，其中令使用第一接口协议的通信去激活的步骤包括如下步骤：

 在共享数据总线上生成一个开始条件信号；以及

- 20 在共享数据总线上传输一个选定的地址，其中该选定的地址没有被任何连接到共享数据总线上的 I²C 设备所使用。

 4. 权利要求 3 的方法，其中在集成电路和第一设备之间间歇地传输数据的步骤包括如下步骤：

 在共享数据总线上生成一个开始条件信号；

 在共享数据总线上传输第一设备的地址；

- 25 通过共享数据总线从集成电路向第一设备发送数据；以及

 在共享数据总线上生成一个结束条件信号。

 5. 权利要求 3 的方法，其中第二接口协议包括串行点对点接口协议。

- 30 6. 权利要求 3 的方法，其中允许使用第二接口协议的通信的步骤包括通过芯片选择线路发送一个第二接口激活信号。

 7. 权利要求 3 的方法，其中第二接口协议包括串行外围接口 (SPI) 协议。

01.07.16

8. 权利要求 3 的方法还包括通过共享时钟线路传输时钟信号的步骤, 该共享时钟线路连接到集成电路、第一设备和第二设备。

9. 权利要求 3 的方法, 其中集成电路包括一个移动电话中的基带控制器。

5 10. 权利要求 9 的方法, 其中第一设备包括一个 LCD 驱动器。

11. 权利要求 3 的方法还包括在完成集成电路和第二设备之间的数据传输之后重新激活使用第一接口协议的通信的步骤。

12. 一种用于传送数据的系统, 包括:

10 具有至少一个数据管脚的集成电路, 该数据管脚与共享数据总线相连, 其中集成电路支持第一接口协议和第二接口协议;

支持第一接口协议的第一驱动器, 该第一驱动器与共享数据总线相连用于使用第一接口协议通过共享数据总线在集成电路和第一设备之间传送数据;

15 支持第二接口协议的第二驱动器, 该第二驱动器与共享数据总线相连用于使用第二接口协议通过共享数据总线在集成电路和第二设备之间传送数据; 以及

20 其中集成电路作用于通过在共享数据总线上发送去激活信号令第一驱动器去激活, 激活第二驱动器来执行集成电路和第二设备之间的通信, 以及在不生成重新激活第一驱动器的重新激活信号的情况下使用第二接口协议向第二设备传输数据。

13. 权利要求 12 的系统还包括连接到集成电路、第一驱动器和第二驱动器的共享时钟总线。

14. 权利要求 12 的系统, 其中第一接口协议包括智能互连通信 (I²C)。

25 15. 权利要求 14 的系统, 其中第二接口协议包括串行点对点接口协议。

16. 权利要求 14 的系统, 其中第二接口协议包括串行外围接口 (SPI) 协议。

30 17. 权利要求 14 的系统, 其中第一驱动器包括一个 LCD 驱动器。

18. 权利要求 14 的系统, 其中集成电路通过发送一个和第一设备无关的地址令第一驱动器去激活。

01.07.16

19. 权利要求 14 的系统，其中集成电路使用芯片选择线路激活第二驱动器。

20. 权利要求 14 的系统，其中集成电路完成向第二设备传输数据之后，通过在共享数据总线上发送一个结束条件来重新激活第一驱动器。

21. 移动电话中的基带控制器，包括：

一个数据管脚，用于通过共享数据总线选择性地传输数据到支持智能互连通信 (I²C) 协议的第一驱动器和支持不同的串行数据接口协议的第二驱动器中的一个；

10 一个芯片选择管脚，用于通过芯片选择线路来传输激活信号以激活第二驱动器的串行数据接口；

一个处理器，用于控制激活信号，当串行数据接口被激活时用于令第一驱动器的 I²C 数据接口去激活，以及用于控制通过共享数据总线的数据传输。

15 22. 权利要求 21 的基带控制器还包括用于向第一驱动器和第二驱动器传输时钟信号的时钟管脚。

23. 权利要求 21 的基带控制器，其中该处理器通过传输和第一驱动器无关的地址令第一驱动器的 I²C 数据接口去激活。

010715

任何传输的耽误。由芯片选择总线 28 上的激活信号控制这种三态条件。因此，当系统 10 正依照串行点对点接口协议进行操作时，芯片选择总线 28 被设置为高，由此允许串行点对点驱动器 18 以它们的正常方式进行操作。另一方面，当系统 10 使用 I²C 接口协议时芯片选择总线 28 被设置为低，使得串行点对点驱动器 18 保持高阻抗以便避免干扰 I²C 通信。不过，本领域的技术人员应当理解，可以颠倒芯片选择总线 28 上的设置或在芯片选择总线 28 上使用其它信号方法来表明何时整个系统 10 正在使用 I²C 接口协议和何时系统 10 正在使用串行点对点（或某种其它）接口协议。

10 现在讨论图 2，其中显示了图 1 说明的电路中时钟总线 24、数据总线 22 以及芯片选择总线 28 上电压电平的表示图。在时间 0 (T=0)，时钟总线 24 上的电压为高时基带控制器 12 通过使得数据总线 22 上的电压由高变低生成一个开始条件。其结果是，连接到基带控制器 12 的每一个 I²C 驱动器 14 开始侦听其唯一地址。在紧接的 7 个时钟周期（即，T=1 到 T=7（没有显示））中传输一个 7-位地址。或者，依据所使用的 I²C 接口类型，可能会发送一个 10-位地址。不过，在这个实例中，由基带控制器 12 传输的地址没有标识任何附属的 I²C 驱动器 14。传输一个未被使用的地址会引起 I²C 驱动器 14 进入非活动状态，由此当所有 I²C 驱动器 14 处于非活动状态时，允许基带控制器 12（或某种其它串行点对点驱动器 18）使用串行点对点接口协议（或某种其它接口协议）来传输数据。

正如前面所提到的，重要的是支持串行点对点接口协议的基带控制器 12 和任何其它设备 18 避免在整个串行点对点通信中无意间生成开始或结束条件。而且，在开始条件和地址数据的传输过程中，串行点对点驱动器 18 应当保持高阻抗以防止对 I²C 通信的任何干扰。在串行点对点通信开始之前，必须通知串行点对点驱动器 18：数据总线 22 可用于串行点对点通信。因此，基带控制器 12 加载芯片选择总线 28，激活接收串行点对点驱动器 18 来接收数据，在这种情形下是开始于 I²C 标准规定的第一地址字节传输之后的第 9 个时钟周期 (T=9)。基带控制器 12 然后传输以及接收驱动器 18 接收，一串二进制数据（即 T=9 时的逻辑 0，T=10 时的逻辑 1，T=11 时的逻辑 0 等等）。

在数据传输完成后，关闭芯片选择总线 28，使得串行点对点驱动

0107.15

器 18 返回到非活动/高阻抗状态。此外，基带控制器 12 在数据传输完成（这里在 $T=15$ ）后的一个时钟周期里生成一个结束条件。结束条件的生成使得附属的 I²C 单元 14 返回到活动状态，在其中它们又开始侦听开始条件（开始于 $T=16$ ）。

5 现在讨论图 3，描述了说明通过 I²C 接口和通过串行点对点接口进行数据传输的信息流和系统操作图。为了启动 I²C 接口上的数据传输，基带控制器 12 在步骤 50 生成一个开始条件，这使得在数据总线 22 上发送一个 I²C 开始信号 52 到所有附属的驱动器 14 或 18。响应该开始信号 52，附属 I²C 驱动器 14 在步骤 54 侦听它们各自的地址。开始于开始条件之后的下一个时钟周期，基带控制器 12 传输一个地址信号 56，该信号包括支持 I²C 接口协议的 LCD 驱动器 14 的唯一地址。其结果是，任何支持 I²C 以及没有被传输的地址所标识的附属设备 14 进入非活动状态，而被寻址到的 LCD 驱动器 14 在数据总线 22 上传输一个应答比特信号 57 以通知基带控制器 12 它已作好接收数据的准备，以及然后在步骤 58 开始侦听数据。

基带控制器 12 开始传输 I²C 数据信号 60。此外，在接收到 I²C 数据信号 60 的每一个字节之后，接收 LCD 驱动器 14 在数据总线 22 上传输一个应答比特信号 62 以通知基带控制器 12：数据已被接收。一旦完成 I²C 数据消息，基带控制器 12 在步骤 64 生成一个结束条件，该条件使得一个结束信号 66 沿着数据总线 22 被传输到所有附属设备 14 和 18。一接收到结束信号 66，所有附属 I²C 单元 14 就立即开始在步骤 68 监视数据总线 22 以等到开始条件。

为启动另一个数据传输（这次使用串行点对点接口协议），基带控制器 12 又一次在步骤 50' 生成一个开始条件，使得开始信号 52' 被转发到所有附属单元 14 和 18。因此，I²C 单元 14 又一次在步骤 54' 开始侦听它们各自的地址。不过，在这种情形下，基带控制器 12 传递一个包含没有被任何附属 I²C 单元 14 使用过的地址的地址信号 70。其结果是，所有附属 I²C 单元 14 在步骤 72 进入非活动状态。然后在步骤 74 基带控制器 12 设置芯片选择线路，向所有附属串行点对点驱动器 18 发送一个芯片选择信号 76。响应该芯片选择信号，在步骤 78 串行点对点驱动器 18 离开高阻抗状态，该状态在 I²C 操作过程为这些驱动器 18 所使用，由此使驱动器 18 能够接收串行点对点数

010715

议中的一个进行通信，该系统应该包括一个最小 I/O 管脚数。这样的方法和系统将允许，例如，使用支持 I²C 的驱动器（当其可得到时），或者如果不能得到 I²C 驱动器或期望有不同性能，则使用串行点对点驱动器。此外，这样的方法和系统将允许集成电路依据与其通信的设备所支持的协议而选择性地使用不同的接口协议。

发明概述

本发明包括用于在集成电路和多个外围设备之间传送数据的方法和系统。该集成电路使用 I²C 接口协议与其中的第一外围设备通信以及使用某种其它接口协议与第二外围设备通信。但是，这两种接口协议都共享同一条数据总线和时钟总线，因此减少了集成电路上所需要的管脚数目。

依据标准的 I²C 接口协议完成集成电路和第一外围设备之间的通信。因此，为了与第一外围设备进行通信，集成电路通过共享数据总线传输一个唯一的 I²C 开始条件，后面带着识别第一外围设备的地址。然后使用该共享数据总线，集成电路传送以及第一外围设备接收预定给第一外围设备的数据。然后第一外围设备在接收到每一个字节之后回复一个应答比特。在数据传输的结尾，传输一个唯一的结束条件。在这样使用 I²C 接口协议的通信过程中，集成电路保持芯片选择总线上的高电压。这个芯片选择总线上的高电压用于通知第二外围设备：正在进行 I²C 通信以及防止第二外围设备妨碍这次通信。

另一方面，在没有进行通过 I²C 接口的通信时，可以开始与第二外围设备之间的通信。在共享数据总线上发送一个结束条件之后，集成电路生成另一个 I²C 开始条件。不过，这次，开始条件后面跟着的不是第一外围设备所用（或者是任何其它附属的 I²C 设备所用）的地址，这将引起第一外围设备（或者是任何其它附属的 I²C 设备）进入非活动状态。然后，假使附属设备中没有一个无意中传输一个可能重新激活附属 I²C 设备的开始或结束条件，则通过移走芯片选择总线上的高电压，激活第二外围设备以及通过共享数据总线可以执行集成电路与第二外围设备之间的通信。一旦完成这样的数据通信，集成电路在共享数据总线上生成一个结束条件，使得附属 I²C 设备再一次开始侦听开始条件。因此，无需要求额外的数据总线和时钟总线，使用其它接口协议的通信可以与 I²C 通信交错，而同时避免两个或更多接口

01.07.16

说明书

接口交错

发明背景

5 发明的技术领域

本发明一般涉及用于在集成电路环境下传送数据的接口，以及特别涉及一种使用多接口在共享总线上传送数据的方法和系统，

相关领域的描述

10 在许多应用中集成电路和其它设备之间的通信使用由飞利浦半导体公司开发的智能互连通信 (I2C)，或 Inter-IC 串行接口。在电信行业，例如，移动电话经常使用 I²C 接口协议进行基带控制器和 LCD 组件之间的通信，基带控制器典型的是一种特定用途集成电路 (ASIC)。移动电话内的其它设备还使用 I²C 接口协议从基带控制器接收数据 (包括命令)，以及将数据传输到基带控制器。

15 不过，由于 I²C 接口受特定的专利保护，LCD 驱动器和其它驱动器的开发商正常地必须得到许可才能够生产支持 I²C 接口协议的驱动器。其结果是，支持 I²C 接口协议的现用的驱动器的供应通常受到限制。随着移动通信行业和使用 I²C 接口协议的其它行业中开发周期不断地缩短，不是总是有时间来定制开发一个支持 I²C 的驱动器。

20 另一方面，现有的现用驱动器中很多支持串行的点对点接口协议，该协议使用四种线路：数据线，时钟线，说明是命令数据或显示

Best Available Copy

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.